

Helpdesk XIMEA

Portal > Knowledgebase > Vision Libraries > OpenCV

OpenCV

Support SK - 2023-10-19 - in Vision Libraries

<https://www.ximea.com/support/wiki/vision-libraries/opencv>

OpenCV



Overview

- [OpenCV](#)
 - [Overview](#)
 - [xiApiPlusOcv API](#)
 - [OpenCV Installation](#)
 - [Sample Application](#)
 - [How to write applications](#)
 - [Documentation](#)
 - [Offline Processing](#)

[OpenCV](#) is a library of programming functions mainly aimed at real-time computer vision, originally developed by Intel. It is free for use under the open source BSD license.

The library is cross-platform and focuses mainly on real-time image processing.

If the library finds Intel's Integrated Performance Primitives on the system, it will use these commercial optimized routines to accelerate itself.

Our cameras support this Vision Library by usage of XIMEA xiApiPlusOcv.

xiApiPlusOcv API

The XIMEA xiApiPlusOcv is an application interface which allows to easily create applications in CPP capable to get images from a camera and use them together with standard OpenCV library.

OpenCV Installation

The sample application for OpenCV Library requires the headers and binaries to be installed before it can be successfully built.

Please follow these guidelines depending on your OS:

- [Windows](#)
- [Linux](#)
- [macOS](#)

Sample Application¶

XIMEA OpenCV sample application captures a sequence of images and displays it on the screen using OpenCV Mat format.

[OpenCV sample code](#) - See this section with various OpenCV examples.

If [XIMEA API Software Package](#) is installed, sample applications with XIMEA OpenCV Library are located:

- **Windows** - in *XIMEA\Examples\XiApiPlusOcv* as a Visual Studio .vcxproj or in *XIMEA\Examples\Sources\XiApiPlusOpenCV*
- **Linux** - in */opt/XIMEA/examples/xiApiPlusOpenCV* directory as a Makefile project or in */opt/XIMEA/samples/xiApiPlusOpenCV*
- **macOS** - in *examples* as a Makefile project

To **run this application**, please follow these steps:

- Download and install [XIMEA API Software Package](#)
- Install OpenCV
- Check the sample application dependencies.

In Visual Studio project settings it is necessary to check OpenCV paths, and also Ximea libraries. Our configuration uses environment variable `$(OPENCV_DIR)` as path where OpenCV is built (e.g. *C:\opencv\build\x64\vc12*).

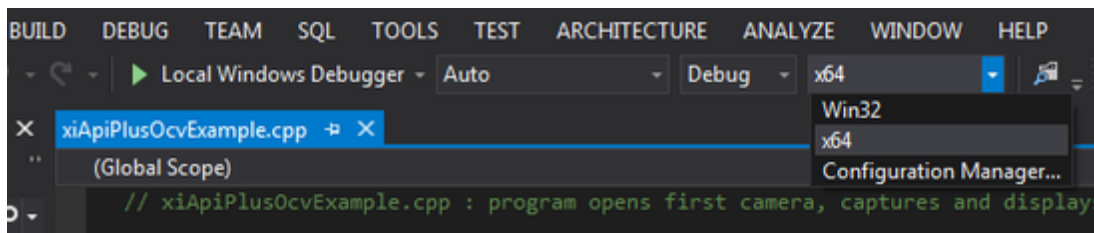
These setting should be checked in the project settings:

- Configuration properties --> C/C++ --> General --> Additional Include Directories
- Configuration properties --> Linker --> General --> Additional Library Directories
- Configuration properties --> Linker --> Input --> Additional Dependencies. The default settings use the *opencv_world310.lib*.
- Build our OpenCV sample application

The Sample application uses the OpenCV *xiApiPlusOcv.hpp* library interface to capture series of frames from the camera and converts each frame to the OpenCV Mat format and shows it on the display.

It uses a **xiAPIplusCameraOcv** class to handle the camera and its features.

When compiling on 64-bit Windows systems the Solution platform in Visual Studio should be set to x64.



How to write applications¹

```
#include <stdio.h>
#include "xiApiPlusOcv.hpp"

using namespace cv;
using namespace std;

int main (void)
{
    try
    {
        xiAPIplusCameraOcv cam;
        cam.OpenFirst();
        cam.SetExposureTime(10000); //10000 us = 10 ms
        cam.StartAcquisition();

        // Read and convert a frame from the camera
        Mat cv_mat_image = cam.GetNextImageOcvMat();
        // Show image on display
        cv::imshow("Image from camera",cv_mat_image);

        cam.StopAcquisition();
        cam.Close();
    }
    catch(xiAPIplus_Exception& exp)
    {
        exp.PrintError(); // report error if some call fails
    }
}
```

The **xiAPIplusCameraOcv** class is used to handle the camera.

In an application, an instance of this class has to be created.

Then the class should be initialized by calling an opening method (e.g. **OpenFirst()**).

Camera parameters can be set by calling **set** methods or read by calling **get** methods.

The camera acquisition starts after **StartAcquisition()** is called.

The captured frames can be read by calling **GetNextImageOcvMat()** - a method that reads a frame from the camera and converts it into OpenCV Mat format that can be directly used for OpenCV processing.

Documentation [1](#)

- XIMEA OpenCV Library is documented at xiApiPlusOcv.hpp file.
- OpenCV is documented in [OpenCV Online C++ Reference](#)

Offline Processing [1](#)

xiApiPlusOcv allows to process already captured and stored images using Offline Processing. Please read more at [xiApiPlusOcv Offline Processing](#).